

INICIAR SESIÓN

NUESTROS PLANES

TODOS LOS
CURSOS

FORMACIONES

CURSOS

PARA
EMPRESAS

ARTÍCULOS DE TECNOLOGÍA > DATA SCIENCE

La diferencia entre las funciones `input()` y `raw_input()` en Python



Yan Orestes

22/10/2020

Las funciones **`input`** y **`raw_input`** pueden traer confusión al programar en **Python**. ¿Dónde usar cada una?

Tengo un programa escrito en Python que necesita de las credenciales de acceso del usuario para funcionar:

```
usuario = input('Ingresa tu nombre de usuario ')\nprint('Hola, ' + usuario)
```

Cuando ejecuto en mi computadora, escribo mi nombre de usuario y recibo una respuesta:

```
Ingresa tu nombre de usuario: yanorestes\nHola, yanorestes
```

¡Exactamente lo que había escrito, capturado por la variable `usuario`!

Llevé el programa a la empresa, pero cuando fui a probarlo, escribí mi nombre de usuario y mira lo que surgió:

```
Ingresa tu nombre de usuario: yanorestes\nTraceback (most recent call last):
```

```
File "login.py", line 1, in <module>
    usuario = input("Ingrese tu nombre de usuario: ")
File "<string>", line 1, in <module>
NameError: name 'yanorestes' is not defined
```

¿Extraño no? ¡*funcionó en mi computadora!* ¿Qué significa este **NameError** que tuvimos como respuesta al ejecutar el código en la computadora de la empresa?

Esta excepción indica que estamos tratando de usar una variable que no estaba definida, en este caso, yanorestes. Pero yanorestes es el inicio de sesión en la string, no es una variable, así que ¿por qué sucede esto?

Revisemos las versiones de Python en cada computadora para ver si tiene algo que ver. Primero, en mi computadora:

```
yan@yan:~> python -V
Python 3.6.4
```

Ahora, en la computadora de la empresa:

```
empresa@xyz:~> python -V
Python 2.7.14
```

¡Ah! Realmente hay una diferencia entre las dos versiones, como sospechábamos.

La función input() en Python 2

¿Por qué nos dio NameError, entonces? La cuestión es que **la función input(), en Python 2, no convierte la entrada del usuario en una string, sino que intenta evaluarla:**

```
cuenta = input('Escriba algo para ejecutar: ')
print(cuenta)
```

Por ejemplo, si enviamos `3 + 5`:

```
Introduzca algo para ser ejecutado: 3 + 5
```

```
8
```

Así que cuando intentamos enviar al usuario de yanorestes al `input()`, lo trata como una variable. Como esta variable no existe, obtenemos `NameError`.

Debido al poco uso de este comportamiento, **fue eliminado en Python 3**. Aún así, podemos simularlo a través de la función de [eval\(\)](#), de esta manera:

```
cuenta = eval(input('Introduzca algo para ser ejecutado: '))
print(cuenta)
```

Entonces, como en el ejemplo anterior obtenemos:

```
Introduzca algo para ser ejecutado: 3 + 5
```

```
8
```

Pero entonces, ¿cómo podemos hacer en **Python 2** lo que la función `input()` hace en **Python 3**?

La función `raw_input()` en Python 2

En **Python 2**, en lugar de la función `input()`, usamos la función `raw_input()` para obtener la entrada del usuario como una string, vea:

```
usuario = raw_input('Ingresa tu nombre de usuario: ')
print('Hola, ' + usuario)
```

Ahora sí:

```
Ingresa tu nombre de usuario: yanorestes
```

```
Hola, yanorestes
```

En **Python 3**, la función `raw_input()` fue simplemente renombrada a `input()`.

¿Qué hacer con las diferentes versiones?

Como habrán notado, hay una inconsistencia aquí, si queremos capturar una entrada de usuario como una string, no hay un solo código que simplemente satisfaga ambas versiones 2 y 3 de Python.

En este punto, nosotros como programadores debemos tomar una decisión pensando en los beneficios de cada versión de Python y especialmente en la mayoría de nuestros usuarios.

"¿Pero qué pasa si necesito que mi programa funcione en cualquiera de las dos versiones?"

Si queremos darle la vuelta a la incompatibilidad, es posible comprobar la versión de Python a través del atributo `version_info.major` del módulo [sys](#), que nos da la posibilidad de crear una condición `if` y seguir el camino correcto, como en el ejemplo:

```
Import sys

if sys.version_info.major == 2:
    usuario = raw_input('Ingresa tu nombre de usuario: ')
elif sys.version_info.major == 3:
    usuario = input('Ingresa tu nombre de usuario: ')
```

El peligro de ejecutar una entrada de usuario

Es muy importante notar lo peligroso que son las funciones `input()` en **Python 2** y `eval()` en ambas versiones.

Ejecutar algo escrito por el usuario abre una tremenda brecha en la seguridad del programa e incluso en la computadora en la que se está usando el programa.

Vea [en este artículo de Ned Batchelder](#) (en inglés) un poco más sobre los peligros del `eval()`. Un simple ejemplo:

```
eval('os.sistema('ls /')')
```

Esto listará los archivos en el directorio raíz de nuestra computadora.

¿Y si en lugar de listar, usamos el comando de borrar? Todos los archivos y directorios de nuestra computadora serían borrados, ¡así de simple!

Por lo tanto, debemos recordar siempre ser cuidadosos al hacer una entrada de usuario.

Conclusión

En este post, aprendimos las diferencias entre las funciones `raw_input()` y `input()`, cómo esto cambia de **Python 2** para **Python 3** e incluso un poco sobre `eval()` y sus peligros.

¿Qué tal aprender más sobre **Python** y sus diversos recursos? Entonces, ¡Mira nuestros cursos de **Python para Data Science** aquí en [Alura](#)!

ARTÍCULOS DE TECNOLOGÍA > DATA SCIENCE

**En Alura encontrarás variados cursos sobre Data Science.
¡Comienza ahora!**

SEMESTRAL

US\$49,90

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas

- ✓ Acceso a todo el contenido de la plataforma por 6 meses

¡QUIERO EMPEZAR A ESTUDIAR!

[Paga en moneda local en los siguientes países](#)

ANUAL

US\$79,90

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

¡QUIERO EMPEZAR A ESTUDIAR!

[Paga en moneda local en los siguientes países](#)

Acceso a todos
los cursos

Estudia las 24 horas,
dónde y cuándo quieras

Nuevos cursos
cada semana

NAVEGACIÓN

PLANES
INSTRUCTORES
BLOG
POLÍTICA DE PRIVACIDAD
TÉRMINOS DE USO
SOBRE NOSOTROS
PREGUNTAS FRECUENTES

¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

BLOG

PROGRAMACIÓN
FRONT END
DATA SCIENCE
INNOVACIÓN Y GESTIÓN
DEVOPS

AOVS Sistemas de Informática S.A
CNPJ 05.555.382/0001-33

SÍGUENOS EN NUESTRAS REDES SOCIALES



ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

CURSOS

Cursos de Programación

Lógica de Programación | Java

Cursos de Front End

HTML y CSS | JavaScript | React

Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

Cursos de DevOps

Docker | Linux

Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics | Liderazgo y Gestión de Equipos | Startups y Emprendimiento